Computational Support for Academic Peer Review: A Perspective from Artificial Intelligence

Simon Price and Peter A. Flach
Department of Computer Science
University of Bristol, Bristol BS8 1UB, UK
{simon.price, peter.flach}@bristol.ac.uk

KEY INSIGHTS

State-of-the-art tools from machine learning and artificial intelligence are making inroads to automate parts of the peer review process; however, many opportunities for further improvement remain.

Profiling, matching and open-world expert finding are key tasks that can be addressed using feature-based representations commonly used in machine learning.

Such streamlining tools also offer perspectives on how the peer review process might be improved: in particular, the idea of profiling naturally leads to a view of peer review being aimed at finding the best publication venue (if any) for a submitted paper.

Creating a more global embedding for the peer review process which transcends individual conferences or conference series by means of persistent reviewer and author profiles is key, in our opinion, to a more robust and less arbitrary peer review process.

1. INTRODUCTION

Peer review is the process by which experts in some discipline comment on the quality of the works of others in that discipline. Peer review of written works is firmly embedded in current academic research practice where it is positioned as the gateway process and quality control mechanism for submissions to conferences, journals and funding bodies across a wide range of disciplines. It is probably safe to assume that peer review in some form will remain a cornerstone of academic practice for years to come, evidence-based criticisms of this process in computer science [33, 32, 45] and other disciplines [28, 23] notwithstanding.

While parts of the academic peer review process have been streamlined in the last few decades to take account of technological advances, there are many more opportunities for computational support that are not currently being exploited. The aim of this article is to identify such opportunities and describe a few early solutions by ourselves and others for automating key stages in the established academic peer review process. When developing these solutions we have found it useful to build on our background in machine learning and artificial intelligence: in particular, we utilise a feature-based perspective in which the hand-crafted features on which conventional peer review usually depends (e.g., keywords) can be improved by feature weighting, selection and construction (see [17] for a broader perspective on the role and importance of features in machine learning).

Twenty-five years ago, at the start of our academic careers, submitting a paper to a conference was a fairly involved and time-consuming process that roughly went as follows. Once an author had produced the manuscript (in the original sense, i.e., manually produced on a typewriter, possibly by someone from the University's pool of typists), he or she would make up to seven photocopies, stick all of them in a large envelope and send them to the program chair of the conference, taking into account that international post would take 3-5 days to arrive. On their end, the program chair would receive all those envelopes, redistribute them to the various members of the program committee, and send them out for review by post in another batch of big envelopes. Reviews would be completed by hand on paper and posted back or brought to the program committee meeting. Finally, notifications and reviews would be sent back by the program chair to the authors by post. Submissions to journals would follow a very similar process.

It is clear that we have moved on quite substantially from this paper-based process – indeed, many of the steps we describe above would seem arcane to our younger readers. These days, papers and reviews are submitted on-line in some *conference management system* (CMS), and all communication is done via e-mail or via message boards on the CMS with all metadata concerning people and papers stored in a database back-end. One could argue that this has made the process much more efficient, to the extent that we now specify the submission deadline up to the second in a particular timezone (rather than approximately as the last post round at the program chair's institution), and can send out hundreds if not thousands of notifications at the touch of a button.

Computer scientists have been studying automated computational support for conference paper assignment since pioneering work in the nineties [14]. A range of methods have been used to reduce the human effort involved in paper allocation, typically with the aim of producing assignments that are similar to the 'gold standard' manual process [13, 16, 34, 18, 9, 37, 30]. Yet, despite many publications on this topic over the intervening years, research results in paper assignment have made relatively few inroads into mainstream CMS tools and everyday peer review practice. Hence, what we have achieved over the last 25 years or so appears to be a *streamlined* process rather than a fundamentally improved one: we

Table 1: A chronological summary of the main activities in peer review, with opportunities for improving the process through computational support.

	Actor	*Activity*	*What can be done now*	*What might be done in future*
I	Author	Paper submission		Recommender systems for publication venue; papers carry full previous reviewing history
II	Program chair	Assembling program committee	Expert finding (Section 4)	PCs for an area rather than a single conference; workload balancing
III	Program chair	Assigning papers for review	Bidding and assignment support (Section 2)	Extending PCs based on submitted papers
IV	Reviewer	Reviewing papers		Advanced reviewing tools that find related work and map the paper under review relative to it
V	Program chair	Discussion and decisions	Reviewer score calibration (Section 3)	More outcome categories; recommender systems for outcomes; more decision time points

believe it would be hard to argue that the decisions taken by program committees today are significantly better in comparison with the paper-based process. But this doesn't mean that opportunities for improving the process don't exist – on the contrary, there is, as we demonstrate in this paper, considerable scope for employing the very techniques that researchers in machine learning and artificial intelligence have been developing over the years.

Table 1 recalls the main steps in the peer review process and highlights current and future opportunities for improving it through advanced computational support. In discussing these it will be helpful to draw a distinction between closed-world and open-world settings. In a *closed-world* setting there is a fixed or pre-determined pool of people or resources. For example, assigning papers for review in a closed-world setting assumes that a program committee or editorial board has already been assembled, and hence the main task is one of matching papers to potential reviewers. In contrast, in an *open-world* setting the task becomes one of finding suitable experts. Similarly, in a closed-world setting an author has already decided which conference or journal to send their paper to, whereas in a open-world setting one could imagine a recommender system that suggests possible publication venues. The distinction between closed and open worlds is gradual rather than absolute: indeed, the availability of a global database of potential publication venues or reviewers with associated metadata would render the distinction one of scale rather than substance. Nevertheless, it is probably fair to say that, in the absence of such global resources, current opportunities tend to be focus on closed-world settings. In the next sections we review work by ourselves and others on steps II, III and V, starting with the latter two which are more of a closed-world nature.

2. ASSIGNING PAPERS FOR REVIEW

In the currently established academic process, peer review of written works depends on appropriate assignment to several expert peers for their review. Identifying the most appropriate set of reviewers for a given submitted paper is a time-consuming and non-trivial task for conference chairs and journal editors – not to mention funding program managers, who rely on peer review for funding decisions. In this section we break the review assignment problem down into its matching and constraint satisfaction constituents, and discuss possibilities for computational support.

Formally, given a set P of papers with $|P| = p$ and a set R of reviewers with $|R| = r$, the goal of paper assignment is to find a binary matrix $A^{r\times p}$ such that $A_{ij} = 1$ indicates that the i-th reviewer has been assigned the j-th paper, and $A_{ij} = 0$ otherwise. The assignment matrix should satisfy various constraints, the most typical of which are: (i) each paper is reviewed by at least c reviewers (typically, $c = 3$); (ii) each reviewer is assigned no more than m papers, where $m = O(pc/r)$; and (iii) reviewers should not be assigned papers for which they have a conflict of interest (this can be represented by a separate binary conflict matrix $C^{r\times p}$). As this problem is underspecified, we will assume that further information is available in the form of a score matrix $M^{r\times p}$ expressing for each paper-reviewer pair how well they are matched by means of a non-negative number (higher means a better match). The best allocation is then the one that maximises the element-wise matrix product $\sum_{i,j} A_{ij}M_{ij}$ while satisfying all constraints [44].

This one-dimensional definition of *best* does not guarantee the *best set* of reviewers if a paper covers multiple topics, e.g. a paper on machine learning and optimisation could be assigned three reviewers who are machine learning experts but none who are optimisation experts. This shortcoming can be addressed by replacing R with the set R^c such that each c-tuple $\in R^c$ represents a possible assignment of c reviewers [25, 24, 42]. Recent works add explicit constraints on topic coverage to incorporate multiple dimensions into the definition of best allocation [31, 26, 40]. Other types of constraints have also been considered, including geographical distribution and fairness of assignments, as have alternative constraint solver algorithms [2, 20, 20, 19, 43]. The score matrix can come from different sources, possibly a combination. In the following sections we review three possible sources: feature-based matching, profile-based matching, and bidding.

2.1 Feature-based matching

To aid assigning submitted papers to reviewers a short list of subject keywords is often required by mainstream CMS tools as part of the submission process, either from a controlled vocabulary, such as the ACM Computing Classification System (CCS)[1], or as a free-text 'folksonomy'. As well as collecting keywords for the submitted papers, taking the further step of also requesting subject keywords from the body of potential reviewers enables CMS tools to make a straightforward match between the papers and the reviewers based on a count of the number of keywords they have in common. For each paper the reviewers can then be ranked in order of the number of matching keywords.

[1] `http://www.acm.org/about/class/`

If the number of keywords associated with each paper and each reviewer is not fixed then the comparison may be normalised by the CMS to avoid overly favouring longer lists of keywords. If the overall vocabulary from which keywords are chosen is small then the concepts they represent will necessarily be broad and likely to result in more matches; conversely, if the vocabulary is large, as in the case of free-text or the ACM CCS, then concepts represented will be finer grained but the number of matches is more likely to be small or even non-existent. Also, manually assigning keywords to define the subject of written material is inherently subjective. In the medical domain, where taxonomic classification schemes are commonplace, it has been demonstrated that different experts, or even the same expert over time, may be inconsistent in their choice of keywords [6, 5].

When a pair of keywords do not literally match, despite having been chosen to refer to the same underlying concept, one technique that is often used to improve matching is to also match their synonyms or syntactic variants – as defined in a thesaurus or dictionary of abbreviations, e.g., treating 'code inspection' and 'walk-through' as equivalent; likewise for 'SVM' and 'support vector machine' or 'λ-calculus' and 'lambda calculus'. However, if such simple equivalence classes are not sufficient to capture important differences between subjects – e.g., if the difference between 'code inspection' and 'walk-through' is significant – then an alternative technique is to exploit the hierarchical structure of a concept taxonomy in order to representation the distance between concepts. In this setting, a match can be based on the common ancestors of concepts – either counting the number of shared ancestors or computing some edge traversal distance between a pair of concepts, e.g., the ACM CCS concept 'D.1.6 Logic Programming' has ancestors 'D.1 Programming Techniques' and 'D. Software', both of which are shared by the concept 'D.1.5 Object-oriented Programming', meaning that D.1.5 and D.1.6 have a non-zero similarity because they have common ancestors.

Obtaining a useful representation of concept similarity from a taxonomy is challenging because the measures tend to assume uniform coverage of the concept space such that the hierarchy is a balanced tree. The approach is further complicated as it is common for certain concepts to appear at multiple places in a hierarchy, i.e. taxonomies may be graphs rather than just trees, and consequently there may be multiple paths between a pair of concepts. The situation grows worse still if different taxonomies are used to describe the subject of written works from different sources because a mapping between the taxonomies is required. Thus it is not surprising that one of the most common findings in the literature on ontology engineering is that ontologies, including taxonomies, thesauri and dictionaries, are difficult to develop, maintain and use [12].

So, even with good CMS support, keyword-based matching still requires manual effort and subjective decisions from authors, reviewers and, sometimes, ontology engineers. One useful aspect of feature-based matching using keywords is that it allows us to turn a heterogeneous matching problem (papers against reviewers) into a homogeneous one (paper keywords against reviewer keywords). Such keywords are thus a simple example of *profiles* that are used to describe relevant entities (papers and reviewers). In the next section we take the idea of profile-based matching a step further by employing a more general notion of profile that incorporates non-feature-based representations such as bags of words.

The Vector Space Model

The canonical task in information retrieval is, given a query in the form of a list of words (terms), to rank a set of text documents D in order of their similarity to the query. In the vector space model, each document $d \in D$ is represented as the multiset of terms (bag-of-words) occurring in that document. The set of distinct terms in D, vocabulary V, defines a vector space with dimensionality $|V|$ and thus each document d is represented as a vector $\vec{d}$ in this space. The query q can also be represented as a vector $\vec{q}$ in this space, assuming it shares vocabulary V. The query and a document are considered similar if the angle θ between their vectors is small. The angle can be conveniently captured by its cosine $\vec{q} \cdot \vec{d} / ||\vec{q}|| \cdot ||\vec{d}||$, giving rise to the *cosine similarity*.

However, if raw term counts are used in vectors $\vec{q}$ and $\vec{d}$ then similarity will: (i) be biased in favour of long documents and; (ii) treat all terms as equally important, irrespective of how commonly they occur across all documents. The *term frequency – inverse document frequency* (tf-idf) weighting scheme compensates for (i) by normalising term counts within a document by the total number of terms in that document, and (ii) by penalising terms that occur in many documents, as follows. The *term frequency* of term t_i in the document d_j is $\mathrm{tf}_{ij} = n_{ij} / \sum_k n_{kj}$. The *inverse document frequency* of term t_i is $\mathrm{idf}_i = \log(|D|/\mathrm{df}_i)$, where *term count* n_{ij} is the number of times term t_i occurs in the document d_j, and *document frequency* df_i of term t_i is the number of documents in D in which term t_i occurs. A term that occurs often in a document has high term frequency; if it occurs rarely in other documents it has high inverse document frequency. The product of the two, tf-idf, thus expresses the extent to which a term characterises a document relative to other documents in D.

2.2 Automatic feature construction with profile-based matching

The main idea of profile-based matching is to automatically build representations of semantically relevant aspects of both papers and reviewers in order to facilitate construction of a score matrix. An obvious choice of such a representation for papers is as a weighted bag-of-words (see sidebar 'The Vector Space Model'). We then need to build similar profiles of reviewers. For this purpose we can represent a reviewer by the collection of all their authored or co-authored papers, as indexed by some online repository such as DBLP [29] or Google Scholar. This collection can be turned into a profile in several ways, including: (i) build the profile from a single document or web page containing the bibliographic details of the reviewer's publications (see SubSift and MLj-Matcher sidebar); or (ii) retrieve or let the reviewer upload full-text of (selected) papers, which are then individually converted into the required representation and collectively averaged to form the profile (see Toronto Paper Matching System (TPMS) sidebar). Once both the papers and the reviewers have been profiled, the score matrix M can be populated with the cosine similarity between the term weight vectors of each paper-reviewer pair.

Profile-based methods for matching papers with reviewers exploit the intuitive idea that the published works of reviewers, in some sense, describe their specific research interests and expertise. By analysing these published works in relation to the body as a whole, discriminating profiles may be produced that effectively characterise reviewer expertise from the content of existing heterogeneous documents ranging from traditional academic papers to web sites, blog posts and social media. Such profiles have applications in their

Experience from SIGKDD'09

Our own experience with bespoke tools to support the research paper review process started when Flach was appointed, with Mohammed Zaki from Rensselaer Polytechnic Institute, program co-chair of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2009 (SIGKDD'09). The initial SubSift tools were written by members of the Bristol Intelligent Systems Laboratory with external collaborators at Microsoft Research Cambridge. As reported in [18] the SubSift tools assisted in the allocation of 537 submitted research papers to 199 reviewers.

Using these tools, each reviewer's bids were initialised using a weighted sum of cosine similarity between the paper's abstract and the reviewer's publication titles as listed in the DBLP computer science online bibliography [29], and the number of shared subject areas. The combined similarity scores were discretised into four bins using manually chosen thresholds, with the first bin being a 0 (no-bid) and the other three being bids of increasing strength: 1 (at a pinch), 2 (willing) and 3 (eager). These initial bids were exported from SubSift and imported into the conference management tool (Microsoft CMT, `cmt.research.microsoft.com`).

Based on the same similarity information, each reviewer was sent an email containing a link to a personalised SubSift generated web page listing details of all 537 papers ordered by initial bid allocation or by either of its two components: keyword matches or similarity to their own published works. The page also listed the keywords extracted from the reviewer's own publications and those from each of the submitted papers. Guided by this personalised perspective, plus the usual titles and abstracts, reviewers affirmed or revised their bids recorded in the conference management tool.

To quantitatively evaluate the performance of the SubSift tools, the bids made by reviewers were considered to be the 'correct assignments' against which SubSift's automated assignments were compared. Disregarding the level of bid, a median of 88.2% of the papers recommended by SubSift were subsequently included in the reviewers' own bids (precision). Furthermore, a median of 80.0% of the papers on which reviewers bid for were ones initially recommended to them by SubSift (recall). Combined, as the harmonic mean of precision and recall, this gives an F-measure of 72.7%. These results suggest that the papers eventually bid on by reviewers were largely drawn from those that were assigned non-zero bids by SubSift. These results on real-world data in a practical setting are comparable with other published results using language models [11, 34, 22].

own right but can also be used to compare one body of documents to another, ranking arbitrary combinations of documents and, by proxy, individuals by their similarity to each other.

From a machine learning point of view, profile-based matching differs from feature-based matching in that the profiles are constructed in a data-driven way without the need to come up with a set of keywords. However, the number of possible terms in a profile can be huge and so systems like TPMS use automatic topic extraction as a form of dimensionality reduction, resulting in profiles with terms chosen from a limited number keywords (topics). As a useful by-product of profiling, each paper and each reviewer is characterised by a ranked list of terms which can be seen as automatically constructed features which could be further exploited, for instance to allocate accepted papers to sessions or to make clear the relative contribution of individual terms to a similarity score (see 'SubSift and MLj Matcher' sidebar).

2.3 Bidding

A relatively recent trend is to transfer some of the paper allocation task downstream to the reviewers themselves, giving them access to the full range of submitted papers and asking them to bid on papers they would like to review. Existing CMS tools offer support for various bidding schemes, including: allocation of a fixed number of 'points' across an arbitrary number of papers; selection of top k papers; rating willingness to review papers according to strength of bid; as well as combinations of these. Hence bidding can be seen as an alternative way to come up with a score matrix that is required for the paper allocation process. There is also the opportunity to register conflicts of interests, if a reviewer's relations with the authors of a particular paper are such that the reviewer is not a suitable reviewer for that paper.

While it is in a reviewer's self-interest to bid, invariably not all reviewers will do so, in which case the papers they are allocated for review may well not be a good match for their expertise and interests. This can be irritating for the reviewer but is particularly frustrating for the authors of the papers concerned. The absence of bids from some reviewers can also reduce the fairness of allocation algorithms in CMS tools [19]. Default options in the bidding process are unable to alleviate this: if the default is 'I cannot review this' the reviewer is effectively excluded from the allocation process, while if the default is to indicate some minimal willingness to review a paper the reviewer is effectively used as a wildcard and will receive those papers that are hardest to allocate.

A hybrid of profile-based matching and manual bidding was explored for the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2009. At bidding time the reviewers were presented with initial bids obtained by matching reviewer publication records on DBLP with paper abstracts (see sidebar 'Experience from SIGKDD'09' for details) as a starting point. Several PC members reported that they considered these bids good enough to relieve them from the temptation to change them, although we feel that there is considerable scope to improve both the quality of recommendations and of the user interface in future work. ICML 2012 further explored the use of a hybrid model and a pre-ranked list of suggested bids[2] . The TPMS software used at ICML 2012 offers other scoring models for combining bids with profile-based expertise assessment [8, 7]. Effective automatic bid initialisation would address the aforementioned problem caused by non-bidding reviewers.

3. REVIEWER SCORE CALIBRATION

Assuming a high-quality paper assignment has been achieved by means of one of the methods described in the previous section, reviewers are now asked to honestly assess the quality and novelty of a paper and its suitability for the chosen venue (conference or journal). There are different ways in which this assessment can be expressed: from a simple yes/no answer to the question 'if it was entirely up to you, would you accept this paper?', via a graded answer on a more common five- or seven-point scale (e.g., Strong Accept (3); Accept (2); Weak Accept (1); Neutral (0); Weak Reject (-1); Reject (-2); Strong Reject (-3)), to graded answers to a set of questions aiming to characterise different aspects of the paper such as novelty, impact, technical quality, and so on.

Such answers require careful interpretation for at least two reasons. The first is that reviewers, and even area chairs, do not have com-

[2]ICML 2012 reviewing – http://hunch.net/?p=2407

Toronto Paper Matching System
The Toronto Paper Matching System TPMS (papermatching.cs.toronto.edu/) originated as a standalone paper assignment recommender for the NIPS 2010 conference and was subsequently loosely integrated with Microsoft's Conference Management Toolkit (CMT) to streamline access to paper submissions for ICML 2012. TPMS requires reviewers to upload a selection of their own papers, reports and other self-selected textual documents which are then analysed to produce their reviewer profile. This places control over the scope of the profile in the hands of the reviewers themselves so that they need only include publications about topics they are prepared to review. Once uploaded, TPMS persists the documents and resultant profile beyond the scope of a single conference, allowing reviewers to reuse the same profile for future conferences, curating their own set of characteristic documents as they see fit.
The scoring model used is similar to the vector-space model but takes a Bayesian approach. In addition, profiles in TPMS can be expressed over a set of hypothesised topics rather than raw terms. Topics are modelled as hidden variables that can be estimated using techniques such as Latent Dirichlet Allocation [3, 7]. This increased expressivity comes at the cost of requiring more training data to stave off the danger of overfitting.

plete information about the full set of submitted papers. This matters in a situation where the total number of papers that can be accepted is limited, as in most conferences (it is less of an issue for journals). The main reason why raw reviewer scores are problematic is that different reviewers tend to use the scale(s) involved in different ways. For example, some reviewers tend to stay to the centre of the scale while others tend to go more for the extremes. In this case it would be advisable to normalise the scores, e.g., by replacing them with z-scores. This corrects for differences in both mean scores and standard deviations among reviewers and is a simple example of *reviewer score calibration*.

In order to estimate a reviewer's score bias (do they tend to err on the accepting side or rather on the rejecting side?) and spread (do they tend to score more or less confidently?) we need a representative sample of papers with a reasonable distribution in quality. For single conferences this is often problematic as the number of papers m reviewed by a single reviewer is too small to be representative, and there can be considerable variation in the quality of papers among different batches which should not be attributed to reviewers. It is however possible to get more information about reviewer bias and confidence by leveraging the fact that papers are reviewed by several reviewers. For SIGKDD'09 we used a generative probabilistic model proposed by colleagues at Microsoft Research Cambridge with latent (unobserved) variables that can be inferred by message-passing techniques such as Expectation Propagation [35]. The latent variables include the true paper quality, the numerical score assigned by the reviewer, and the thresholds this particular reviewer uses to convert the numerical score to the observed recommendation on the seven-point scale. The calibration process is described in more detail in [18].

An interesting manifestation of reviewer variance came to light through an experiment with NIPS reviewing in 2014 [27]. The PC chairs decided to have one-tenth (166) of the submitted papers reviewed twice, each by three reviewers and one area chair. It turned out that the accept/reject recommendations of the two area chairs differed in about a quarter of the cases (43). Given an overall acceptance rate of 22.5%, roughly 38 of the 166 double-reviewed papers were accepted following the recommendation of one of the area chairs; about 22 of these would have been rejected if the recommendation of the other area chair had been followed instead (assuming the disagreements were uniformly distributed over the two possibilities), which suggests that more than half (57%) of the accepted papers would not have made it to the conference if reviewed a second time.

What can be concluded from what came to be known as the 'NIPS experiment' beyond these basic numbers is up for debate. It is worth pointing out that, while the peer review process eventually leads to a binary accept/reject decision, paper quality most certainly isn't: while a certain fraction of papers clearly deserves to be accepted, and another fraction clearly deserves to be rejected, the remaining papers have pros and cons which can be weighted up in different ways. So if two reviewers assign different scores to papers this doesn't mean that one of them is wrong, but rather that they picked up on different aspects of the paper in different ways.

We suggest that a good way forward is to think of the reviewer's job to 'profile' the paper in terms of its strong and weak points, and separate the reviewing job proper from the eventual accept/reject decision. One could imagine a situation where a submitted paper could go to a number of venues (including the 'null' venue), and the reviewing task is to help decide which of these venues is the most appropriate one. This would turn the peer review process into a matching process, where publication venues have a distinct profile (whether it accepts theoretical or applied papers, whether it puts more value on novelty or on technical depth, etc.) to be matched by the submission's profile as decided by the peer review process. Indeed, some conferences already have a separate journal track which implies some form of reviewing process to decide which venue is the most suitable one.[3]

4. ASSEMBLING PEER REVIEW PANELS

The formation of a pool of reviewers, whether for conferences, journals or funding competitions, is a non-trivial process that seeks to balance a range of objective and subjective factors. In practice, the actual process by which a program chair assembles a program committee varies from, at one extreme, inviting friends and co-authors plus their friends and co-authors, through to the other extreme of a formalised election and representation mechanism. The current generation CMSs do not offer computational support for the formation of a balanced program committee; they assume prior existence of the list of potential reviewers and instead concentrate on supporting the administrative workflow of issuing and accepting invitations.

4.1 Expert Finding

This lack of tool support is surprising considering the body of relevant work in the long-established field of *expert finding* [47, 1, 15, 34, 11]. Over the years since the first Text Retrieval Conference (TREC) in 1992, the task of finding experts on a particular topic has featured regularly in this long-running conference series and is now an active subfield of the broader text information retrieval discipline. Expert finding has a degree of overlap with

[3]For example, the *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)* has a journal track where accepted papers are presented at the conference but published either in the *Machine Learning* journal or in *Data Mining and Knowledge Discovery*.

SubSift and MLj-Matcher
SubSift, short for 'submission sifting', was originally developed to support paper assignment at SIGKDD'09 and subsequently generalised into a family of web services and re-usable web tools (`www.simsift.com`). The submission sifting tool composes several SubSift web services into a workflow driven by a wizard-like user interface that takes the Program Chair through a series of web forms of a paper-reviewer profiling and matching process. On the first form, a list of PC member names is entered. SubSift looks up these names on DBLP and suggests author pages which, after any required disambiguation, are used as documents to profile the PC members. Behind the scenes, beginning from a list of bookmarks (urls), SubSift's harvester robot fetches one or more DBLP pages per author, extracts all publication titles from each page and aggregates them into a single document per author. In the next form, the conference paper abstracts are uploaded as a CSV file and their text is used to profile the papers. After matching PC member profiles against paper profiles, SubSift produces reports with ranked lists of papers per reviewer, and ranked lists of reviewers per paper. Optionally, by manually specifying threshold similarity scores or by specifying absolute quantities, a CSV file can be downloaded with initial bid assignments for upload into a CMS.
For the Editor-in-Chief of a journal, the task of assigning a paper to a member of the editorial board for their review can be viewed as a special case of the conference paper assignment problem (without bidding), where the emphasis is on finding the best match for one or a few papers. We built an alternative user interface to SubSift that supports paper assignment for journals. Known as *MLj Matcher* in its original incarnation, this tool has been used since 2010 to support paper assignment for the *Machine Learning* journal as well as other journals.

the fields of *bibliometrics*, the quantitative analysis of academic publications and other research-related literature [38, 21], and *scientometrics*, which extends the scope to include grants, patents, discoveries, data outputs and, in the UK, more abstract concepts such as 'impact' [4]. Expert finding tends to be more profile-based (e.g., based on the text of documents) than link-based (e.g., based on cross-references between documents) although content analysis is an active area of bibliometrics in particular and has been used in combination with citation properties to link research topics to specific authors [11]. Even though by comparison with bibliometrics, scientometrics encompasses additional measures, in practice the dominant approach in both domains is citation analysis of academic literature. Citation analysis measures the properties of networks of citation amongst publications and has much in common with hyperlink analysis on the web, where these measures employ similar graph theoretic methods designed to model reputation, with notable examples including 'Hubs and Authorities' and PageRank. Citation graph analysis, using a particle-swarm algorithm, has been used to suggest potential reviewers for a paper on the premise that the subject of a paper is characterised by the authors it cites [39].

Harvard's Profiles Research Network Software (RNS)[4] exploits both graph-based and text-based methods. By mining high-quality bibliographic metadata from sources like PubMed, Profiles RNS infers implicit networks based on keywords, co-authors, department, location and similar researchers. Researchers can also define their own explicit networks and curate their list of keywords and publications. Profiles RNS supports expert finding via a rich set of searching and browsing functions for traversing these networks. Profiles RNS is a noteworthy open source example of a growing body of *research intelligence* tools that compete to provide definitive databases of academics that, while varying in scope, scale and features, collectively constitute a valuable resource for a program chair seeking new reviewers. Well-known examples include free-to-use sites like academia.edu, Google Scholar, Mendeley, Microsoft Academic Search, ResearchGate and numerous others that mine public data or solicit data directly from researchers themselves, as well as pay-to-use offerings like Elsevier's Reviewer Finder.

4.2 Data Issues

There is a wealth of publicly available data about the expertise of researchers that could, in principle, be used to profile program committee members (without requiring them to choose keywords or upload papers) or to suggest a ranked list of candidate invitees for any given set of topics. Obvious data sources include academic home pages, online bibliographies, grant awards, job titles, research group membership, events attended as well as membership of professional bodies and other reviewer pools. Despite the availability of such data, there are a number of problems in using it for the purpose of finding an expert on a particular topic.

If the data is to be located and used automatically then it is necessary to identify the individual or individuals described by the data. Unfortunately a person's name is not guaranteed to be a unique identifier (UID): often not being globally unique in the first place, they can also be changed through title, choice, marriage and so on. Matters are made worse because many academic reference styles use abbreviated forms of a name using initials. International variations in word ordering, character sets and alternative spellings make name resolution even more challenging for a peer review tool. Indeed, the problem of author disambiguation is sufficiently challenging to have merited the investment of considerable research effort over the years, which has in turn led to practical tool development in areas with similar requirements to finding potential peer reviewers. For instance, Profiles RNS supports finding researchers with specific expertise and includes an Author Disambiguation Engine using factors such as name permutations, email address, institution affiliations, known co-authors, journal titles, subject areas and keywords. To address these problems in their own record systems, publishers and bibliographic databases like DBLP and Google Scholar have developed their own proprietary UID schemes for identifying contributors to published works. However, there is now considerable momentum behind the non-proprietary Open Researcher and Contributor ID (ORCID)[5] and publishers are increasingly mapping their own UIDs onto ORCID UIDs. A subtle problem remains for peer review tools when associating data, particularly academic publications, with an individual researcher because a great deal of academic work is attributed to multiple contributors. Hope for resolving individual contributions comes from a concerted effort to better document all outputs of research, including not only papers but also websites, datasets and software, through richer metadata descriptions of *Research Objects* [10].

4.3 Balance and Coverage

Finding candidate reviewers is only part of a program chair's task in forming a committee – attention must also be paid to coverage and balance. It is important to ensure that more popular areas get proportionately more coverage than less popular ones whilst also not excluding less well known but potentially important new areas.

[4] `http://profiles.catalyst.harvard.edu`

[5] `http://orcid.org`

There is thus a subjective element to balance and coverage that is not entirely captured by the score matrix. Recent work seeks to address this for conferences by refining clusters, computed from a score matrix, using a form of crowdsourcing from the program committee and from the authors of accepted papers [**?**] Another example of computational support for assembling a balanced set of reviewers comes not from conferences but from a US funding agency, the National Science Foundation (NSF).

The NSF presides over a budget of over $7 billion (FY 2015) and receives 40,000 proposals per year, with large competitions attracting 500-1500 proposals; peer review is part of the NSF's core business. Around a decade ago, the NSF developed Revaide, a data mining tool to help them find proposal reviewers and to build panels with expertise appropriate to the subjects of received proposals [22]. In constructing profiles of potential reviewers the NSF decided against using bibliographic databases like Citeseer or Google Scholar, for the same reasons we discussed in Section 4.2. Instead they took a closed-world approach by restricting the set of potential reviewers to authors of past (single-author) proposals that had been judged 'fundable' by the review process. This ensured the availability of a UID for each author and reliable metadata, including the author's name and institution, which facilitated conflict of interest detection. Reviewer profiles were constructed from the text of their past proposal documents (including references and resumes) as a vector of the top 20 terms with the highest tf-idf scores. Such documents were known to be all of similar length and style, which improved the relevance of the resultant tf-idf scores. The same is also true of the proposals to be reviewed and so profiles of the same type were constructed for these.

For a machine learning researcher, an obvious next step towards forming panels with appropriate coverage for the topics of the submissions would be to cluster the profiles of received proposals and use the resultant clusters as the basis for panels, for example matching potential reviewers against a prototypical member of the cluster. Indeed, prior to Revaide the NSF had experimented with the use of automated clustering for panel formation but those attempts had proved unsuccessful for a number of reasons: the sizes of clusters tended to be uneven; clusters exhibited poor stability as new proposals arrived incrementally; there was a lack of alignment of panels with the NSF organisational structure; and, similarly, no alignment with specific competition goals, such as increasing participation of under-represented groups or creating results of interest to industry. So, eschewing clustering, Revaide instead supported the established manual process by annotating each proposal with its top 20 terms as a practical alternative to manually-supplied keywords.

Other ideas for tool support in panel formation were considered. Inspired by conference peer review, NSF experimented with bidding but found that reviewers had strong preferences towards well-known researchers and this approach failed to ensure that there were reviewers from all contributing disciplines of a multidisciplinary proposal – a particular concern for NSF. Again, manual processes won out. However, Revaide did find a valuable role for clustering techniques as a way of checking manual assignments of proposals to panels. To do this Revaide calculated an "average" vector for each panel, by taking the central point of the vectors of its panel members, and then compared each proposal's vector against every panel. If a proposal's assigned panel is not its closest panel then the program director is warned. Using this method, Revaide proposed better assignments for 5% of all proposals. Using the same representation, Revaide was also used to classify orphaned proposals, suggesting a suitable panel. Although the classifier was only 80% accurate, which is clearly not good enough for a fully automated assignment, it played a valuable role within the NSF workflow: so, instead of each program director having to sift through, say, 1,000 orphaned proposals they received an initial assignment of, say, 100 of which they would need to reassign around 20 to other panels.

5. CONCLUSIONS AND OUTLOOK

We have demonstrated that state-of-the-art tools from machine learning and artificial intelligence are making inroads to automate and improve parts of the peer review process. Allocating papers (or grant proposals) to reviewers is an area where much progress has been made. The combinatorial allocation problem can easily be solved once we have a score matrix assessing for each paper-reviewer pair how well they are matched.[6] We have described a range of techniques from information retrieval and machine learning that can produce such a score matrix. The notion of profiles (of reviewers as well as papers) is useful here as it turns a heterogeneous matching problem into a homogeneous one. Such profiles can be formulated against a fixed vocabulary (bag-of-words) or against a small set of topics. Although it is fashionable in machine learning to treat such topics as latent variables that can be learned from data, we have found stability issues with latent topic models (i.e., adding a few documents to a collection can completely change the learned topics) and have started to experiment with hand-crafted topics (e.g., encyclopaedia or Wikipedia entries) which extend keywords by allowing their own bag-of-words representations.

A perhaps less commonly studied area where nevertheless progress has been achieved concerns interpretation and calibration of the intermediate output of the peer reviewing process: the aspects of the reviews that feed into the decision making process. In their simplest form these are scores on an ordinal scale, that are often simply averaged. However, averaging assessments from different assessors – which is common in other areas as well, e.g., academic marking – is fraught with difficulties as it makes the unrealistic assumption that each assessor scores on the same scale. It is possible to adjust for differences between individual reviewers, particularly when a reviewing history is available that spans multiple conferences. Such a global reviewing system which builds up persistent reviewer (and author) profiles is something that we support in principle, although many details need to be worked out before this is viable.

We also believe that it would be beneficial if the role of individual reviewers shifted away from being an *ersatz* judge attempting to answer the question 'would you accept this paper if it was entirely up to you?' towards a more constructive role of characterising – and indeed, profiling – the paper under submission. Put differently, besides suggestions for improvement to the authors, the reviewers attempt to collect metadata about the paper that is used further down the pipeline to decide the most suitable publication venue. In principle this would make it feasible to decouple the reviewing process from individual venues, something that would also enable better load balancing and scaling [46]. In such a system, authors and reviewers would be members of some central organisation which has the authority to assign papers to multiple publication venues – a futuristic scenario, perhaps, but it is worth thinking about the peculiar constraints that our current conference- and journal-driven system imposes, and which clearly leads to a sub-optimal situation

[6]This holds for the simple version stated in Section 2, but further constraints might complicate the allocation problem.

in many respects.

The computational methods we described in this article have been used to support other academic processes outside of peer review, including a personalised conference planner app for delegates[7], an organisational profiler [36] and a personalised course recommender for students based on their academic profile [41]. The table in the introductory section lists a few other possible future directions for computation support of academic peer review itself. We hope that they, and this article, stimulate our readers to think about ways in which the academic peer review process – this strange dance in which we all participate in one way or another – can be future-proofed in a sustainable and scalable way.

6. REFERENCES

[1] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 43–50, New York, NY, USA, 2006. ACM.

[2] S. Benferhat and J. Lang. Conference paper assignment. *International Journal of Intelligent Systems*, 16(10):1183–1192, 2001.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[4] L. Bornmann, B. Bowman, J. Bauer, W. Marx, H. Schier, and M. Palzenberger. Standards for using bibliometrics in the evaluation of research institutes. *Next Generation Metrics*, 2013.

[5] A. A. Boxwala, M. Dierks, M. Keenan, S. Jackson, R. Hanscom, D. W. Bates, and L. Sato. Review paper: Organization and representation of patient safety data: Current status and issues around generalizability and scalability. *Journal of the American Medical Informatics Association: JAMIA*, 11(6):468–478, 2004.

[6] J. Brixey, T. Johnson, and J. Zhang. Evaluating a medical error taxonomy. *Proceedings of the American Medical Informatics Association (AMIA) Symposium*, 2002.

[7] L. Charlin and R. Zemel. The toronto paper matching system: An automated paper-reviewer assignment system. In *ICML Workshop on Peer Reviewing and Publishing Models*, 2013.

[8] L. Charlin, R. Zemel, and C. Boutilier. A framework for optimizing paper matching. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 86–95, Corvallis, Oregon, 2011. AUAI Press.

[9] L. Charlin, R. S. Zemel, and C. Boutilier. A framework for optimizing paper matching. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

[10] D. De Roure. Towards computational research objects. In *Proceedings of the 1st International Workshop on Digital Preservation of Research Methods and Artefacts*, DPRMA '13, pages 16–19, New York, NY, USA, 2013. ACM.

[11] H. Deng, I. King, and M. R. Lyu. Formal models for expert finding on DBLP bibliography data. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 163–172, Washington, DC, USA, 2008. IEEE Computer Society.

[12] V. Devedzić. Understanding ontological engineering. *Commun. ACM*, 45(4):136–144, Apr. 2002.

[13] N. Di Mauro, T. Basile, and S. Ferilli. Grape: An expert review assignment component for scientific conference management systems. In M. Ali and F. Esposito, editors, *Innovations in Applied Artificial Intelligence*, volume 3533 of *Lecture Notes in Computer Science*, pages 789–798. Springer Berlin Heidelberg, 2005.

[14] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 233–244, New York, NY, USA, 1992. ACM.

[15] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 418–430, Berlin, Heidelberg, 2007. Springer-Verlag.

[16] S. Ferilli, N. Di Mauro, T. Basile, F. Esposito, and M. Biba. Automatic topics identification for reviewer assignment. In M. Ali and R. Dapoigny, editors, *Advances in Applied Artificial Intelligence*, volume 4031 of *Lecture Notes in Computer Science*, pages 721–730. Springer Berlin Heidelberg, 2006.

[17] P. Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, 2012.

[18] P. A. Flach, S. Spiegler, B. Golénia, S. Price, J. G. R. Herbrich, T. Graepel, and M. J. Zaki. Novel tools to streamline the conference review process: Experiences from SIGKDD'09. *SIGKDD Explorations*, 11(2):63–67, December 2009.

[19] N. Garg, T. Kavitha, A. Kumar, K. Mehlhorn, and J. Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, Sept. 2010.

[20] J. Goldsmith and R. H. Sloan. The ai conference paper assignment problem. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 2007.

[21] S. Harnad. Open access scientometrics and the uk research assessment exercise. *Scientometrics*, 79(1):147–156, April 2009.

[22] S. Hettich and M. J. Pazzani. Mining for proposal reviewers: lessons learned at the national science foundation. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 862–871, New York, NY, USA, 2006. ACM.

[23] C. Jennings. Quality and value: The true purpose of peer review. *Nature*, 2006.

[24] M. Karimzadehgan and C. Zhai. Integer linear programming for constrained multi-aspect committee review assignment. *Inf. Process. Manage.*, 48(4):725–740, July 2012.

[25] M. Karimzadehgan, C. Zhai, and G. Belford. Multi-aspect expertise matching for review assignment. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1113–1122, New York, NY, USA, 2008. ACM.

[26] N. M. Kou, L. H. U., N. Mamoulis, and Z. Gong. Weighted coverage based reviewer assignment. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 2031–2046, New York, NY, USA, 2015. ACM.

[27] J. Langford and M. Guzdial. The arbitrariness of reviews, and advice for school administrators. *Commun. ACM*, 58(4):12–13, Mar. 2015.

[7] `http://www.ecmlpkdd2012.net/attending/apps/`

[28] P. A. Lawrence. The politics of publication. *Nature*, (422):259–261, March 2003.

[29] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, SPIRE 2002, pages 1–10, London, UK, 2002. Springer-Verlag.

[30] X. Liu, T. Suel, and N. Memon. A robust model for paper reviewer assignment. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 25–32, New York, NY, USA, 2014. ACM.

[31] C. Long, R. C. Wong, Y. Peng, and L. Ye. On good and fair paper-reviewer assignment. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 1145–1150, 2013.

[32] K. Mehlhorn, M. Y. Vardi, and M. Herbstritt. Publication culture in computing research (dagstuhl perspectives workshop 12452). *Dagstuhl Reports*, 2(11), 2013.

[33] B. Meyer, C. Choppy, J. Staunstrup, and J. van Leeuwen. Viewpoint: Research evaluation for computer science. *Commun. ACM*, 52(4):31–34, Apr. 2009.

[34] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 500–509, New York, NY, USA, 2007. ACM.

[35] T. Minka. Expectation propagation for approximate bayesian inference. In J. S. Breese and D. Koller, editors, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann, 2001.

[36] S. Price and P. A. Flach. Mining and mapping the research landscape. In *Digital Research Conference*. University of Oxford, September 2013.

[37] S. Price, P. A. Flach, S. Spiegler, C. Bailey, and N. Rogers. SubSift web services and workflows for profiling and comparing scientists and their published works. *Future Generation Comp. Syst.*, 29(2):569–581, 2013.

[38] A. Pritchard et al. Statistical bibliography or bibliometrics. *Journal of documentation*, 25(4):348–349, 1969.

[39] M. A. Rodriguez and J. Bollen. An algorithm to determine peer-reviewers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 319–328, New York, NY, USA, 2008. ACM.

[40] N. D. Sidiropoulos and E. Tsakonas. Signal processing and optimization tools for conference review and session assignment. *IEEE Signal Process. Mag.*, 32(3):141–155, 2015.

[41] A. Surpatean, E. N. Smirnov, and N. Manie. Master orientation tool. In L. D. Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, editors, *ECAI*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 995–996. IOS Press, 2012.

[42] W. Tang, J. Tang, T. Lei, C. Tan, B. Gao, and T. Li. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, January 2012.

[43] W. Tang, J. Tang, and C. Tan. Expertise matching via constraint-based optimization. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 34–41, Washington, DC, USA, 2010. IEEE Computer Society.

[44] C. J. Taylor. On the optimal assignment of conference papers to reviewers. Technical Report MS-CIS-08-30, Computer and Information Science Department, University of Pennsylvania, 2008.

[45] D. Terry. Publish now, judge later. *Commun. ACM*, 57(1):44–46, 2014.

[46] M. Y. Vardi. Scalable conferences. *Commun. ACM*, 57(1):5–5, Jan. 2014.

[47] D. Yimam-seid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce*, 13:2003, 2003.